Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method of improving performance in a Java computer application program executable by a Java virtual machine (JVM), comprising the steps of:

obtaining information associated with garbage collection; [[and]]

deducing changes in performance that will result from modifying the Java computer application program; and

modifying the Java computer application program, wherein a cost of garbage collection to program performance of the Java computer application program is estimated using a duration of an average garbage collection event and a frequency of garbage collection events.

2. (Cancelled)

3. (Currently Amended) The method of claim [[2]] 1, wherein the cost of garbage collection is reduced by reducing either or both of the duration and frequency.

4. (Currently Amended) The method of claim [[2]] 1, wherein the duration depends on an amount of garbage that must be cleaned up, an algorithm used to do the collecting or copying, a heap compaction, a cost of reconciling object references that are moved, and a number of finalizers that must be executed.

5. (Currently Amended) The method of claim [[2]] 1, wherein the frequency depends on the rate of object creation, the heap fragmentation, the size of the heap, and the garbage collection policy.

6. (Currently Amended) The method of claim 1, wherein the Java computer application program is changed by reducing memory from a footprint of the Java computer application program.

7. (Original) The method of claim 6, wherein given the amount of memory to be reduced from the footprint, a total duration for a run, and how much of the run is spent in garbage collection, the number of additional transactions that can be computed during the run is determined.

8. (Original) The method of claim 7, wherein the information associated with garbage collection is obtained from a verbosegc.

9. (Cancelled)

10. (Currently Amended) A computer system capable of running a Java application program by a Java virtual machine (JVM), comprising:

a garbage heap associated with garbage collection events, wherein garbage collection events have an average duration and frequency;

instructions for estimating changes in performance resulting from modification of the Java application program using information obtained about the garbage collection events; and

instructions for modifying the Java application program, wherein a cost of garbage collection to program performance of the Java application program is estimated using a duration of an average garbage collection event and a frequency of garbage collection events.

11. (Cancelled)

12. (Currently Amended) The system of claim [[11]] 10, wherein the duration depends on an amount of garbage that must be cleaned up, an algorithm used to do the collecting or copying, a heap compaction, a cost of reconciling object references that are moved, and a number of finalizers that must be executed.

13. (Currently Amended) The system of claim [[11]] 10, wherein the frequency depends on the rate of object creation, the heap fragmentation, the size of the heap, and the garbage collection policy.

14. (Currently Amended) The method of claim 10, wherein the Java ~~computer~~ application program is changed by deducting memory from a footprint of the Java ~~computer~~ application program.

15. (Original) The method of claim 14, wherein given the amount of memory to be deducted from the footprint, a total duration for a run, and how much of the run is spent in garbage collection, the number of additional transactions that can be computed during the run is determined.

16. (Original) The method of claim 15, wherein the information associated with garbage collection is obtained from a verbosegc.

17. (Currently Amended) A computer program product in a computer readable medium for improving performance in a Java computer application program executable by a Java virtual machine (JVM), comprising ~~the steps of~~:

first instructions for obtaining information associated with garbage collection; and

second instructions for deducing changes in performance that will result from modifying the Java computer application program, wherein a cost of garbage collection to program performance of the Java computer application program is estimated using a duration of an average garbage collection event and a frequency of garbage collection events;

wherein the Java computer program is changed by deducting memory from a footprint of the Java computer application program.

18. (Cancelled)

19. (Currently Amended) The method of claim [[18]] 17, wherein the cost of garbage collection is reduced by reducing either or both of the duration and frequency.

20. (Currently Amended) The method of claim [[18]] 17, wherein the duration depends on an amount of garbage that must be cleaned up, an algorithm used to do the collecting or copying, a heap compaction, a cost of reconciling object references that are moved, and a number of finalizers that must be executed.

21. (Currently Amended) The method of claim [[18]] 17, wherein the frequency depends on the rate of object creation, the heap fragmentation, the size of the heap, and the garbage collection policy.

22. (Original) The method of claim 17, wherein given the amount of memory to be deducted from the footprint, a total duration for a run, and how much of the run is spent in garbage collection, the number of additional transactions that can be computed during the run is determined.

23. (Original) The method of claim 22, wherein the information associated with garbage collection is obtained from a verbosegc.